# A Novel Approach to Model Generation for Heterogeneous Data Classification

**Rong Jin*, Huan Liu†**
*Dept. of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824
rongjin@cse.msu.edu
† Department of Computer Science and Engineering, Arizona State University, Tempe, AZ85287-8809
hliu@asu.edu

## Abstract

Ensemble methods such as bagging and boosting have been successfully applied to classification problems. Two important issues associated with an ensemble approach are: how to generate models to construct an ensemble, and how to combine them for classification. In this paper, we focus on the problem of model generation for *heterogeneous data classification*. If we could partition heterogeneous data into a number of homogeneous partitions, we will likely generate reliable and accurate classification models over the homogeneous partitions. We examine different ways of forming homogeneous subsets and propose a novel method that allows a data point to be assigned multiple times in order to generate homogeneous partitions for ensemble learning. We present the details of the new algorithm and empirical studies over the UCI benchmark datasets and datasets of image classification, and show that the proposed approach is effective for heterogeneous data classification.

## 1 Introduction

Ensemble approaches such as bagging and boosting have been successfully applied to many classification problems [Dietterich, 2000; Bauer and Kohavi, 1999]. The basic idea of ensemble methods is to construct a number of classifiers over training data and then classify new data points by taking a (weighted) vote of their predictions. Thus, two important issues associated with an ensemble approach are: 1) how to generate *accurate* yet *diverse* classification models, and 2) how to combine the models for ensemble classification. Diverse classifiers ensure good ensembles [Quinlan, 1996]. In this paper, we focus on the first issue with an emphasis on *heterogeneous data classification*. Heterogeneous data classification refers to the problem when input data of a single class are widely distributed into multiple modes. It arises when training data are collected under different environments or through different sources. An example of heterogeneous data classification is image classification, in which labeled images are acquired from multiple resources and exhibit disparate characteristics. For instance, some images are black and white, and others are colorful.

A widely used approach for constructing an ensemble of models is to sample different subsets from the training data and create a classification model for each subset. Bagging [Briemann, 1996] and AdaBoost [Schapire and Singer, 1999] are two representative methods in this category. Bagging randomly draws samples from the training data with replacement and AdaBoost samples training data according to a dynamically changed distribution, which is updated by putting more weight on the misclassified examples and smaller weights on the correctly classified examples. Clearly, both methods do not treat homogeneous data and heterogeneous data differently.

For ensemble methods to work effectively on heterogeneous data, one intuitive solution is to first divide the heterogeneous data into a set of homogeneous partitions and then to create a model for each partition of data. Member classifiers built with different homogeneous partitions will likely result in good diversity of an ensemble. One way to realize this homogeneity-based partition is to employ standard clustering algorithms, such as K-means [Hartigan and Wong, 1979] and the EM clustering algorithm [Celeux and Govaert, 1992]. An example is the Gaussian Mixture Model (GMM). But, in general, there are two problems with this simple clustering approach:

- *Single cluster membership*. Most clustering algorithms assume that cluster membership is mutually exclusive and each data point can only belong to a single cluster. Even though the EM clustering algorithm allows soft membership for a data point, in the resulting clusters, each data point still only belongs to a single cluster [Witten and Frank, 2000]. Therefore, when we use these clustering algorithms to partition data, if the number of clusters is large and the subsets of training data formed by a clustering algorithm are mutually disjoint, some clusters may have a very small number of data points, which can lead to unreliable classification models. This is similar to the data fragmentation problem occurred in decision tree induction [Quinlan, 1993]. In contrast, the subsets of training data produced by Bagging and AdaBoost are not mutually disjoint. For example, in bootstrap sampling, each subset contains around 63.2% of the original training data.
- *Unbalanced cluster sizes*. Since most clustering algorithms do not have control over cluster sizes, unbalanced cluster sizes resulting from clustering cannot be easily cor-

rected. When the resulting clusters have very different sizes, a classifier built over a small cluster can be unreliable and thus degrade the performance of the ensemble in forming final ensemble classification. On the contrary, both Bagging and AdaBoost have data samples of similar sizes when learning different models. Note that there have been previous efforts on balancing the sizes of different clusters, particularly for spectral clustering algorithms (e.g., the normalized cut algorithm [Melta & Shi, 2001]). But, since the control of cluster size comes indirectly from the objective function, the resulting clusters can still have unbalanced sizes.

In sum, a clustering approach may produce homogeneous data partitions, but cannot ensure similar sizes of different partitions; methods like Bagging can produce equally sized partitions, but partitions are not homogeneous. Therefore, we need a novel approach to partitioning data into homogeneous subsets of similar sizes in ensemble learning for heterogeneous data classification.

The goal of this work is to divide heterogeneous data into homogeneous subsets of similar sizes in order to generate reliable and accurate classification models. By focusing on homogeneous subsets, we do not require that each data point belong to one subset; by ensuring similar sizes of data subsets, each classification model can be built with a similar number of data points. In this paper, we propose a HISS (Homogeneous data In Similar Size) algorithm specially designed for the above purposes for heterogeneous data classification. Specifically, HISS allows the user to specify the size of a subset. For example, the user can ask the algorithm to create 20 subsets with each containing 40% of the original data. This algorithm is similar to the bootstrap sampling procedure in that both the number of subsets and the percentage of training data covered by each cluster can be specified and varied. However, it differs from the simple bootstrap sampling procedure in that it puts the similar data points into a single subset while bootstrap sampling randomly selects data to form a subset. This property is important in ensemble learning for classifying heterogeneous data. We will use *strata* for the *homogeneous data partitions*, and *subsets* for data partitions resulting from random sampling.

## 2 Related Work

There have been many previous studies on how to create an ensemble of models. The methods for constructing an ensemble of models can be categorized into five groups [Dietterich, 2000]: 1) *Bayesian methods*, which creates an ensemble of model by sampling them from a estimated posterior model distribution; 2) *Sampling training examples*, which creates multiple subsets of training examples and trains a classifier for each of the subsets; 3) *Sampling input features*, which creates a number of subsets of the input features and a classifier is built for each subset of input features; 4) *Error correct output code (ECOC)*, which convert a multiple class problem into a set of binary class problems; 5) *Injecting randomness*, that generates ensembles of classifiers by injecting randomness into the learning algorithm.

Among the five categories, our work is closely related to the second one, which creates multiple classifiers by sampling training examples. Important methods in this group include Bagging [Brieman, 1996] and AdaBoost [Schapire and Singer, 1999]. Although these methods have been shown to be effective for classification, they are not designed to take into account characteristics of heterogeneous data. In this paper, we propose HISS –an algorithm that constructs homogeneous strata from heterogeneous data while maintains the nice property of boostrap sampling procedure - each stratum contains a similar number of data points.

Another line of research closely related to this work is the study of clustering algorithms. In general, clustering algorithms can be categorized into parametric approaches and non-parametric approaches. The parametric approach is to find a parametric model that minimizes a cost function associated with instance-cluster assignments. Such methods include the Mixture Model [Celeux and Govaert, 1992] and K-means algorithm. For the non-parametric approaches, a cost function is minimized by either merging two separate clusters into a larger one or dividing a cluster into two smaller ones. The representative examples of this category are the agglomerative approach and the divisive approach.

Most clustering approaches assume that each data point only belongs to a single cluster. This assumption may not be appropriate since the ultimate goal of clustering is to group similar data points together. When it is uncertain to assign a data point to a single cluster, it is better off assigning it to multiple clusters. Although the traditional probabilistic model and the fuzzy clustering algorithm allow for multi- or soft-memberships, the uncertainty of cluster membership is only exploited during the process of estimation. In the resulting clusters, each data point is assigned to only a single cluster. Furthermore, most clustering algorithms do not have any control over the size of clusters. Hence, the resulting clusters can be very unbalanced in size and the clusters of too small sizes could be useless in learning.

## 3 The HISS Algorithm for Model Generation

### 3.1 From Probabilistic Clustering to HISS

We first describe the traditional probabilistic clustering algorithm, and then introduce algorithm HISS.

The general idea of probabilistic clustering is to describe data with a mixture of generative models. Optimal parameters are usually obtained by maximizing the likelihood of data using the mixture model. Let $n$ be the number of input data points, $K$ be the number clusters, $\{x_1, x_2, ..., x_n\}$ be the input data, and $\{m_1, m_2, ..., m_K\}$ be the underlying models that generate the data. By assuming that each data point is generated from a mixture of models $\{m_1, m_2, ..., m_K\}$, we have the likelihood of the data written as:

$$l(\{m_i\}_{j=1}^{K}, \boldsymbol{\tau}) = \sum_{i=1}^{n} \log\left( \sum_{j=1}^{K} \tau_i^j \, p(x_i \mid m_j) \right) \qquad (1)$$

where $p(x_i \mid m_j)$ is the likelihood of generating $x_i$ from the model $m_j$, and $\tau_i^j$ is the likelihood for data point $x_i$ to be in the j-th cluster. Based on the assumption that each data point can only belong to a single cluster, we have constraint $\sum_{j=1}^{K} \tau_i^j = 1$. An example of probabilistic clustering is the Gaussian Mixture Model (GMM), in which both $\tau_i^j$ and $p(x_i \mid m_j)$ are parameterized as:

$$\tau_i^j = \theta_j, \text{ and}$$
$$p(x_i \mid m_j) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_j \mid^{1/2}} \exp\left( -\frac{(x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)}{2} \right) \quad (2)$$

where $\theta_j$ denotes the prior for the j-th cluster, and $\mu_j$ and $\Sigma_j$ are the mean and variance matrix for the j-th cluster, respectively. Expectation and Maximization algorithm (EM) (Dempster et al, 1977) can be used to search for the optimal parameters.

By removing the constraint $\sum_{j=1}^{K} \tau_i^j = 1$, we allow each data point to belong to multiple homogeneous clusters, or in short, strata. Hence, the optimization problem becomes

$$\begin{aligned} &\max_{m_i, \tau_i^j} l(\{m_i\}_{j=1}^K, \boldsymbol{\tau}) = \sum_{i=1}^{n} \log\left( \sum_{j=1}^{K} \tau_i^j \, p(x_i \mid m_j) \right) \\ &\text{subject to} \\ &0 \le \tau_i^j \le 1 \text{ for } i = 1, ..., n, \text{ and, } j = 1, ..., K \end{aligned} \quad (3)$$

where all $\tau_i^j$ are constrained to between 0 and 1 to maintain the probability interpretation. It is easy to see that the optimal solution is to set all $\tau_i^j$ to be 1, which means that each data point is included in every stratum.

To avoid the trivial solution for $\tau_i^j$, we choose to enforce the percentage of training data that are covered by each cluster to be a predefined constant $\gamma$, i.e.,

$$\frac{1}{n} \sum_{i=1}^{n} \tau_i^j = \gamma, \; 0 \le \gamma \le 1, \text{ for } j = 1, ..., K \quad (4)$$

With the above constraint, we guarantee that the number of data points that support each stratum is around $\gamma n$.

Compared to the single membership constraint, this new constraint has the following two advantages: 1) ***It does not assume that each data point has to belong to one stratum***. For this new stratifying method, on average each data point can belong to $\gamma K$ number of strata. Therefore, when $\gamma K$ is larger than one, each data point is allowed to be in more than one stratum simultaneously. 2) ***It ensures that different strata have balanced numbers of data points***. In contrast to most clustering algorithms, the new algorithm ensures almost the same size for each stratum. This is particularly important to the research goal of this paper - generating a reliable and accurate ensemble for heterogeneous data. By setting $\gamma$ to be a reasonably large value (0.4 in this work), we ensure that each stratum has a sufficiently large number of examples for building a statistical learning model. For later reference, we refer this new clustering approach as "**HISS**", which stands for Homogeneous data In Similar Size.

### 3.2 Optimization for HISS

Putting Equations (3) and (4) together, we have:

$$\begin{aligned} &\max_{m_i, \tau_i^j} l(\{m_i\}_{j=1}^K, \boldsymbol{\tau}) = \sum_{i=1}^{n} \log\left( \sum_{j=1}^{K} \tau_i^j \, p(x_i \mid m_j) \right) \\ &\text{subject to} \\ &\frac{1}{n} \sum_{i=1}^{n} \tau_i^j = \gamma, \; 0 \le \gamma \le 1, \text{ for } j = 1, ..., K \\ &0 \le \tau_i^j \le 1 \text{ for } i = 1, ..., n, \text{ and, } j = 1, ..., K \end{aligned} \quad (5)$$

Let us assume the Gaussian distribution for $p(x_i \mid m_j)$, i.e., $p(x \mid m_j) \sim N(\mu_j, \sigma_j)$. Following the idea of the EM algorithm, the difference in the likelihood of data between two consecutive iterations is bound by:

$$\begin{aligned} &l(\{m_i(t+1)\}_{i=1}^K, \boldsymbol{\tau}(t+1)) - l(\{m_i(t)\}_{i=1}^K, \boldsymbol{\tau}(t)) \\ &\ge \sum_{i=1}^{n} \sum_{j=1}^{K} \upsilon_i^j(t) \log\left[ \frac{\tau_i^j(t+1)}{\tau_i^j(t)} \right] + \\ &\sum_{i=1}^{n} \sum_{j=1}^{K} \upsilon_i^j(t) \log\left[ \frac{p(x_i \mid m_j(t+1))}{p(x_i \mid m_j(t))} \right] \end{aligned} \quad (6)$$

where $\upsilon_i^j$ is defined as

$$\upsilon_i^j(t) = \frac{\tau_i^j(t) \, p(x_i \mid m_j(t))}{\sum_{j=1}^{K} \tau_i^k(t) \, p(x_i \mid m_k(t))} \quad (7)$$

Thus, the optimal solutions for the mean and variance of Gaussian distribution can be obtained as follows:

$$\mu_j(t+1) = \frac{\sum_{i=1}^{n} \upsilon_i^j(t) x_i}{\sum_{i=1}^{n} \upsilon_i^j(t)}, \sigma_j^2(t+1) = \frac{\sum_{i=1}^{n} \upsilon_i^j(t) x_i^2}{\sum_{i=1}^{n} \upsilon_i^j(t)} - \mu_j^2(t+1)$$

However, the optimal solution for $\tau_i^j$ is rather difficult to obtain because of the inequality constraints $0 \le \tau_i^j \le 1$. Directly optimizing the Equation (6) with only the equality constraint will result in the following solution for $\tau_i^j(t+1)$:

$$\tau_i^j(t+1) = \frac{\upsilon_i^j(t) \gamma n}{\sum_{i=1}^{n} \upsilon_i^j(t)} \quad (8)$$

Apparently, the above solution will always be nonnegative if $\tau_i^j(t)$ is nonnegative. However, it does not guarantee that $\tau_i^j(t+1)$ is not greater than 1.

**Finding Optimal** $\tau_i^j(t+1)$

**Inputs**: $\upsilon_i^j(t)$ for $i=1,...,n$ and $j=1,...,K$

**Outputs**: $\tau_i^j(t+1)$ that maximizes Equation (7).

**Initialization**:

$\tau_i^j(t+1)=0$ for $i=1,...,n$ and $j=1,...,K$

**for** each cluster $j$

**do**

For all examples $i$,

set $\tau_i^j(t+1)=1$ if $\tau_i^j(t+1)>1$

Compute the probability mass

$$s=\gamma n-\left|\{i \mid \tau_i^j(t+1)=1\}\right|$$

Re-compute

$$\tau_i^j(t+1)=s\frac{\upsilon_i^j(t)}{\sum_{\{i|\tau_i^j(t+1)<1\}}\upsilon_i^j(t)} \quad \forall j \text{ s.t. } \tau_i^j(t+1)<1$$

**while** ($\exists j$ s.t. $\tau_i^j(t+1)>1$)

**end**

**Figure 1**: Algorithm for finding optimal $\tau_i^j(t+1)$

In order to satisfy the inequality constraints $0\le\tau_i^j\le 1$, we use the KKT conditions [Fletcher, 1987] to efficiently adjust the value of $\tau_i^j(t+1)$. The basic idea is to reset $\tau_i^j$ to be 1 whenever the output from Equation (10) violates the constraint $0\le\tau_i^j\le 1$. After the adjustment, we will re-compute $\tau_i^j(t+1)$ that are less than 1 using Equation (8). The procedure of adjusting and recomputing $\tau_i^j(t+1)$ will continue until no $\tau_i^j(t+1)$ violates the constraint. Figure 1 shows the detailed steps for finding the optimal solution for $\tau_i^j(t+1)$. Due to the space limit, the proof for the optimality of the algorithm in Figure 1 is not provided here.

### 3.3 Classifying Heterogeneous Data

For classification problems, heterogeneous data can be found in many applications and in experiments:

1) *Data acquired from multiple sources*. In many cases, training data are acquired from multiple sources. Because each source has its own data distribution that may be different from others, the data merged from multiple sources are therefore heterogeneous. For example, consider building a classification model for outdoor scenes. The training images are collected from several different types of videos. Some of the videos are news stories and some of them are of advertisement. Some of them are of high quality and some of them are not. Thus, the widely disparate characteristics in videos cause the merged data to be heterogeneous.

2) *Data by converting a multiple class problem into a set of binary class problems*. In order to apply the binary class classification algorithm to multiple class case, we need to convert the classification problem of multiple classes into a set of binary class problems. The representative examples include the one-against-all approach and error correct output coding (ECOC) method [Dietterich, 1995]. During this process, multiple classes are grouped into two subsets of classes. Data points from one subset of classes are used as positive examples and the remaining are used as negative examples. Because both the positive and negative pools can be comprised of examples from multiple classes, it will create data heterogeneity for each of the binary classes.

| Data Set | # Examples | #Class | # Features |
|---|---|---|---|
| Ecoli | 327 | 5 | 7 |
| Pendigit | 2000 | 10 | 16 |
| Glass | 204 | 5 | 10 |
| Yeast | 1479 | 10 | 8 |
| Vehicle | 946 | 4 | 17 |
| Image/Indoor | 3500 | 2 | 190 |
| Image/Outdoor | 3500 | 2 | 126 |

**Table 1**: Description of datasets for the experiment for heterogeneous data classification.

As discussed, an intuitive solution to classifying heterogeneous data is to create a set of classification models with each classifier built on a homogeneous partition (stratum) of the data, and then combine classifiers for the final prediction. The traditional clustering algorithms are not designed for this task because of the potential unbalanced cluster-sizes and the data fragmentation problem. With the proposed algorithm HISS, we can avoid these two problems by setting the parameter to be large (0.4 in the experiment).

In sum, to classify heterogeneous data, we first apply HISS to obtain homogeneous strata and then create a classification model for each stratum to form an ensemble. We will refer to this model generation method as 'HISS-based Model Generation' in our empirical study next. Finally, a stacking approach [Wolpert, 1992] is used to combine models that are generated by the HISS-based model generation method for the final prediction of the ensemble.

## 4. Experimental Study

The experimental study is designed to answer the following questions:

1) *Is the proposed model generation method effective for classifying heterogeneous data*? To this end, we compare the proposed model generation method to Bagging and AdaBoost in classifying heterogeneous datasets.

2) *Is the proposed HISS algorithm effective for generating reliable models*? To address this question, we will apply both the proposed HISS algorithm and the probabilistic clustering algorithm to partition the training data and build a classification model for each partition.

### 4.1 Experimental Design

Seven different datasets are used in the experiments: five multiple class datasets from the UCI Machine Learning repository [Blake and Merz, 1998] and two binary class data-

| Data Set | Baseline | AdaBoost (Standard) | Bagging (Standard) | HISS-based Ensemble | Bagging (Stacking) | AdaBoost (Stacking) |
|---|---|---|---|---|---|---|
| Ecoli | 0.047 (0.012) | 0.046 (0.006) | 0.057 (0.014) | 0.037 (0.006) | 0.046 (0.006) | 0.059 (0.006) |
| Pendigit | 0.010 (0.003) | 0.013 (0.003) | 0.012 (0.002) | 0.008 (0.002) | 0.012 (0.001) | 0.012 (0.003) |
| Glass | 0.382 (0.027) | 0.385 (0.081) | 0.379 (0.046) | 0.161 (0.044) | 0.379 (0.027) | 0.379 (0.027) |
| Yeast | 0.314 (0.012) | 0.320 (0.023) | 0.313 (0.013) | 0.313 (0.013) | 0.315 (0.012) | 0.315 (0.008) |
| Vehicle | 0.103 (0.020) | 0.163 (0.048) | 0.131 (0.024) | 0.048 (0.012) | 0.100 (0.017) | 0.085 (0.033) |
| Image/Indoor | 0.153 (0.008) | 0.140 (0.007) | 0.156 (0.014) | 0.140 (0.013) | 0.157 (0.011) | 0.144 (0.007) |
| Image/Outdoor | 0.116 (0.008) | 0.111 (0.017) | 0.120 (0.011) | 0.088 (0.005) | 0.114 (0.006) | 0.112 (0.007) |

**Table 2**: Classification errors for the baseline model (SVM), AdaBoost, Bagging and the propose model generation method ('HISS-based Ensemble'). The column 'Bagging (Stacking)' refers to the case when the ensemble of models is created by the Bagging algorithm but combined through the stacking approach using an SVM. The same is for the column 'AdaBoost (Stacking)'. The variance of classification error is listed in parenthesis.

sets for image classification. The characteristics of these seven datasets are listed in Table 1.

For the multiple class datasets, we introduce the heterogeneity into the data by converting the original multiple-class problem into a binary one. Similar to the one-against-all approach, examples from the most popular class are used as the positive instances and examples from the remaining classes are assigned to the negative class. Because data of the negative class are from multiple classes, we would expect some degree of heterogeneity inside the negative class. For the two datasets of image classification, they both are binary classification problems. The heterogeneity of data is due to the fact that images are from seven different video clips and each video clip provides 500 images. Since each video clip is of different type (e.g., varied quality in images), we would expect certain amount of heterogeneity within the data.

The baseline algorithm used in this experiment is support vector machine [Burger, 1998]. In all the experiments, each ensemble method generates 20 different SVMs; a stacking approach [Wolpert, 1992] that also uses a SVM is employed to combine the outputs from all 20 models to form the final prediction of the ensemble. For each experiment, we randomly select 70% of the data as training and the remaining 30% as testing. The experiment is repeated 10 times and the average classification error of the ten runs is used as the final result with the variance of classification errors.

## 4.2 Heterogeneous Data Classification

Table 2 shows classification errors for the baseline support vector machine, the proposed HISS-based ensemble learning approach, standard Bagging and standard AdaBoost. First, we can see that the baseline model performs well comparing with both standard Bagging and AdaBoost. This observation indicates that these seven heterogeneous datasets are rather difficult for the standard ensemble approaches to learn. In contrast, the proposed HISS-based ensemble method performs better than the baseline model and the two standard ensemble methods. For the datasets 'Glass', 'Vehicle', and 'Image/Outdoor', the improvement is substantial, from 38.2% to 16.1% for 'Galss', 10.3% to 4.8% for 'Vehicle', and from 11.6% to 8.8% for 'Image/Outdoor'.

| Data Set | HISS | EM (3 Clusters) | EM (10 Clusters) |
|---|---|---|---|
| Ecoli | 0.037(0.006) | 0.448 (0.021) | 0.448 (0.021) |
| Pendigit | 0.008(0.002) | 0.081 (0.043) | 0.110 (0.023) |
| Glass | 0.161(0.044) | 0.292 (0.101) | 0.353 (0.017) |
| Yeast | 0.313 (0.013) | 0.314 (0.013) | 0.314 (0.019) |
| Vehicle | 0.048 (0.012) | 0.219 (0.068) | 0.052 (0.026) |
| Image/Indoor | 0.140(0.013) | 0.184 (0.022) | 0.203 (0.014) |
| Image/Outdoor | 0.088(0.005) | 0.156 (0.031) | 0.182 (0.036) |

**Table 3**: Classification error for using different clustering algorithms for model generation. 'EM' refers to using Expectation-Maximization algorithm to cluster data.

Since the HISS-based ensemble method uses the stacking approach for combining different models, it is different from the combination method that is used by AdaBoost and Bagging. To address this difference, we conduct the experiments that apply a stacking method to combine the models generated by both Bagging and AdaBoost. The results are listed in Table 2 on the right side of the HISS-based approach, titled as 'Bagging (Stacking)' and 'AdaBoost (Stacking)', respectively. Compared these results to the results of 'Bagging (Standard)' and 'AdaBoost (Standard)', we see that there is no substantial change in classification errors when using a stacking approach to combine models in ensemble learning. For all the seven datasets, the ensemble of models generated by HISS performs the best. The reason why a stacking approach is useful for the HISS-based model generation method but not to the other two is that models generated by the HISS-based algorithm are much more diverse than the ones generated by both Bagging and AdaBoost. As a result, applying another layer of classification model to combine the outputs from the distinguishable models (or stacking) will be able to take full advantage of all the models and obtain the best performance.

Based on the above discussion, we conclude that the HISS-based ensemble model is more effective for classifying heterogeneous data than existing ensemble approaches.

## 4.3 Comparison with Other Clustering-based Ensemble Methods

The advantage of HISS versus the traditional clustering algorithms is that HISS allows each data point to be in multiple different strata. Thus it can ensure that the number of

data points distributed over each stratum is of similar size and sufficiently large.

In this experiment, we use both the traditional clustering algorithm and the proposed HISS algorithm for model generation and see how different they are in classifying the heterogeneous datasets. To observe the effect due to the trade-off between the number of strata and the number of data points in each stratum, we consider two different numbers of strata (or clusters) for the traditional clustering algorithm: 10 and 3($\approx 1/\gamma$). We did not use 20 clusters in the comparison because for some datasets the traditional clustering algorithm is unable to produce the full twenty clusters. The traditional clustering algorithm used in the experiment is the probabilistic EM clustering algorithm. Similar to the HISS-based ensemble approach, a stacking method is used to combine models generated by the EM clustering algorithm. The results for using EM clustering algorithms for model construction are listed in Table 3, titled 'EM (3 clusters)' and 'EM (10 clusters)'. As suggested by Table 3, the increasing number of clusters can lead to degraded performance. This is because a large number of clusters will form clusters with a small number of data points, which can be insufficient for building a reliable classification model. On the other hand, as already indicated in the previous study [Ditterich, 2000], being able to generate a relatively large number of models is critical to the success of the ensemble approach. The proposed HISS algorithm can satisfy both needs by introducing the substantial overlapping between different clusters. As shown in Table 3, the HISS-based method outperforms the EM_clustering-based ensemble approaches substantially for almost all datasets except for 'Yeast' (similar). The most noticeable cases are 'Ecoli' and 'Pendigit', for which the classification errors of EM-based clustering approaches are one order more than that of the HISS-based ensemble algorithm.

Based on the above experiments and analysis, we conclude that the HISS-based model generation is an effective method for model generation in ensemble learning for heterogeneous data classification.

## 5. Conclusion and Future Work

In this paper, we propose and examine a new method for generating an ensemble of models, which is to first partition data into homogeneous subsets and then create a model for each subset. A traditional clustering algorithm like EM is not suitable for the task of partitioning data due to potential size-unbalanced clusters and the data fragmentation problem. To address these two problems, we propose a novel algorithm HISS, which allows for data overlapping between different clusters (strata) and promises size-balanced clusters. Empirical studies over seven different heterogeneous datasets have shown that this new HISS-based model generation method performs very well for heterogeneous data classification. Currently, the proposed HISS algorithm assumes equal size for each stratum (cluster). One possible extension is to examine alternatives to balance sizes of clusters. For example, instead of enforcing all the clusters to have one size, we can constrain the sizes of the clusters into a specified range to allow some flexibility in maintaining high homogeneity of clusters.

## References


[Hartigan and Wong, 1979] Hartigan, J.A. and Wong, M.A., A K-means Clustering Algorithm, Applied Statistics 28: 100-108

[Briemann, 1996] Briemann, L., Bagging Predicator, Machine Learning 26, 123-140, 1996

[Schapire and Singer, 1999]Schapire, R.E, and Singer, Y., Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3): 291-336, 1999

[Celeux and Govaert, 1992] Celeux, G. and Govaert, G., A Classification EM Algorithm for Clustering and Two Stochastic Versions, Computational Statistics & Data Analysis, vol. 14, pp. 315-332, 1992

[Dietterich, 2000] Dietterich, T.G., Ensemble Methods in Machine Learning. In Multiple Classier Systems, Cagliari, Italy, 2000

[Dempster et al., 1977] Dempster, A.P., Laird, N.M., and Rubin, D.B., Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal statistical Society, Series B, 39(1): 1-38, 1977

[Fletcher, 1987] Fletcher, R., Practical Methods of Optimization. John Wiley and Sons, Inc., 2nd edition, 1987.

[Dietterich and Bakiri, 1995] Dietterich, T.G. and Bakiri, G., Solving Multiclass Learning Problems via Error-Correcting Output Codes. Journal of Artificial Intelligence Research, (2):263–286, 1995.

[Wolpert, 1992] Wolpert, D.H., Stacked Generalization. Neural Networks, 5:241-259, Pergamon Press, 1992.

[Burges, 1998] Burges, C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2):955-974, 1998.

[Witten and Frank, 2000] Witten, I.H. and Frank, E., Data Mining: Practical Machine Learning Tools with Java Implementations. Morgan Kaufmann, 2000.

[Blake and Merz, 1998] Blake, C. and Merz, C., UCI repository of ma-chine learning databases. http://www.ics.uci.edu/mlearn/MLRepository.html.

[Quinlan, 1993] Quinlan, R.J., C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, 1993.

[Quinlan, 1996] Quinlan R.J., Bagging, boosting, and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI 96), 1996.

[Shi and Malik, 2000] Shi, J., and Malik, J., Normalized Cut and Image Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888-905, 2000

[Bauer and Kohavi, 1999] Bauer, E. and Kohavi, R., An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, 36(1):105-139, 1999.